

Proyecto de Gestión de Publicaciones

Este proyecto es una aplicación web desarrollada con **Next.js** y **Axios** para gestionar publicaciones. Los usuarios pueden agregar, editar y ver publicaciones en un formato de tarjeta interactivo. La aplicación también incluye una funcionalidad para alternar entre mostrar todos los posts o solo una parte de ellos. Los datos se obtienen de la API pública de [JSONPlaceholder](#).

Características

- **Lista de publicaciones:** Muestra una lista de publicaciones obtenidas de la API.
- **Crear nueva publicación:** Permite a los usuarios agregar nuevas publicaciones mediante un formulario.
- **Editar publicaciones:** Los usuarios pueden editar el título y el contenido de las publicaciones existentes.
- **Eliminar publicaciones:** Los usuarios pueden eliminar publicaciones de las publicaciones existentes.
- **Visibilidad de publicaciones:** Los usuarios pueden alternar entre ver solo las primeras 4 publicaciones o todas las disponibles.
- **Manejo de errores:** Se muestra un mensaje de error si ocurre un problema al cargar, agregar o editar publicaciones.
- **Interfaz interactiva:** Utiliza botones de acción y una interfaz de usuario basada en tarjetas.
- **Lista de comentarios:** Muestra una lista de comentarios obtenidas de la API y filtradas por el id de la publicación.
- **Eliminar comentarios:** Los usuarios pueden eliminar comentarios de los comentarios existentes.

Instalación

1. Clona el repositorio:

```
git clone https://github.com/Saimon1520/prueba-jr.git
```

2. Navega a la carpeta del proyecto:

```
cd prueba-jr
```

3. Instala las dependencias:

```
npm install
```

4. Ejecuta el proyecto:

```
npm run dev
```

5. Abre la aplicación en tu navegador: <http://localhost:3000>

Estructura del Proyecto

- **src/app/layout.tsx**: Componente principal que maneja la estructura global del sitio, incluyendo la barra de navegación y el diseño base.
- **src/app/page.tsx**: Página de inicio donde se muestra un mensaje de bienvenida y el formulario para agregar publicaciones.
- **src/app/posts/page.tsx**: Página que muestra una lista de publicaciones con la opción de editar y eliminar cada una.
- **src/app/components/Navbar.tsx**: Componente de navegación que permite la navegación entre las páginas principales del sitio.
- **src/app/components/PostForm.tsx**: Formulario para agregar o editar publicaciones.
- **src/app/context/PostContext.tsx**: Contexto global que maneja el estado de las publicaciones y proporciona funciones para agregar, editar y alternar la visibilidad de las publicaciones.
- **src/app/styles/globals.css**: Estilos globales de la aplicación, incluyendo soporte para un modo oscuro.
- **tsconfig.json**: Configuración de TypeScript.
- **package.json**: Dependencias y scripts del proyecto.
- **src/app/posts/[id]/page.tsx**: Página que muestra una lista de comentarios con la opción eliminar cada uno.
- **src/app/context/CommentContext.tsx**: Contexto global que maneja el estado de los comentarios y proporciona la función de eliminar los comentarios.

Funcionalidades Detalladas

Cargar Publicaciones

- Las publicaciones se cargan al inicio mediante una solicitud GET a la API de **JSONPlaceholder**.

Agregar Publicaciones

- Los usuarios pueden agregar nuevas publicaciones mediante el formulario en la página de inicio.

Editar Publicaciones

- Los usuarios pueden editar una publicación haciendo clic en el botón "Editar" en cada tarjeta de publicación.

Alternar Visibilidad de Publicaciones

- Los usuarios pueden alternar entre ver solo las primeras 4 publicaciones o todas las disponibles con el botón "Show More" o "Show Less".

Dependencias

- **Next.js**: Framework de React para aplicaciones de servidor.
- **React**: Librería para construir interfaces de usuario.
- **Axios**: Cliente HTTP para realizar solicitudes.
- **Tailwind CSS**: Framework de CSS para estilos rápidos y responsivos.

Desarrollo

- **Instalación de dependencias:**

```
npm install
```

- **Ejecutar el proyecto:**

```
npm run dev
```

- **Lint:** El proyecto usa ESLint para el análisis de código.

```
npm run lint
```

Recomendaciones Propuestas

1. Implementar un login simple

Recomiendo desarrollar un sistema de inicio de sesión básico en el que el usuario ingrese su correo electrónico. Esto permitirá filtrar las publicaciones de manera más efectiva y personalizada según el usuario.

2. Llevar el proyecto a producción y conectar con una base de datos

Recomiendo llevar el proyecto a un entorno de producción y conectarlo a una base de datos real. Esto permitirá hacer modificaciones reales y trabajar con datos auténticos, mejorando la funcionalidad y escalabilidad del sistema.

3. Hacer la página web compatible con modo claro y oscuro

Recomiendo implementar una funcionalidad que permita alternar entre el modo claro y el modo oscuro en la interfaz de usuario, adaptando los colores del diseño según la preferencia del usuario para mejorar la experiencia visual.

4. Incorporar la sección de álbumes de JSONPlaceholder

Recomiendo agregar la sección de álbumes disponible en la API de JSONPlaceholder para enriquecer el contenido de la página y proporcionar más recursos interactivos para los usuarios.

Decisiones Técnicas Tomadas

1. Decidí crear un contexto para las publicaciones y otro para los comentarios debido a que la API no permite realizar cambios reales en los datos. Esto me permite manejar localmente los cambios realizados, como editar, agregar o eliminar publicaciones, y eliminar comentarios. Sin embargo, mantuve las consultas a la API, ya que comprendí que era importante seguir consumiéndola como parte del proyecto.

2. Opté por implementar un navbar para facilitar la navegación dentro de la página web. Esto también sirve como una representación de cómo podría extenderse la página en el futuro, permitiendo agregar más apartados según sea necesario.
3. Elegí usar la licencia MIT, ya que planeo hacer público el proyecto una vez que sea revisado por la empresa que actualmente me está evaluando.
4. Decidí utilizar Axios como cliente HTTP porque ofrece una mejor gestión de errores (como los códigos 404, 400 y 500), es compatible con navegadores antiguos y maneja automáticamente las respuestas en formato JSON, lo que simplifica el desarrollo.

Licencia

Este proyecto está bajo la Licencia MIT. Consulta el archivo LICENSE.txt para más detalles.